АНАЛИЗ ТЕКСТОВЫХ ДАННЫХ, С ЦЕЛЬЮ ВЫДЕЛЕНИЯ ЗНАЧИМЫХ ФРАГМЕНТОВ ОБ АНАЛИЗАХ ПАЦИЕНТОВ

Д. Д. Богданов, С.В. Аксёнов
Томский политехнический университет
ddb4@tpu.ru

Введение

За последние несколько лет информационные системы стали неотъемлемой частью современного мира. Информационные системы (ИС) используются во многих областях, основанных на информации той или иной области, Например: в Бухгалтерском учете, Документообороте, в медицине и т.д.

Информационной системой называется комплекс, включающий вычислительное и коммуникационное оборудование, программное обеспечение, лингвистические средства и информационные ресурсы, а также системный персонал и обеспечивающий поддержку динамической информационной модели некоторой части реального мира для удовлетворения информационных потребностей пользователей [1].

У каждой клиники есть свои лаборатории по анализу крови. В этих лабораториях есть приборы, которые анализируют кровь. В данные приборы помещается пробирки с кровью, которые в дальнейшем подвергаются тщательному анализу. После анализа прибор выдает справку об анализе крови в электронном виде. Далее эту справку можно распечатать и выдать врачу или пациенту. Наилучшим вариантом считается, что лучше работать с медицинскими справками в электронном виде. Во-первых, потому что это экономит такой ресурс, как бумага, во-вторых, это сэкономит много места в архивах, где хранятся медицинские карты пациентов, в-третьих все эти электронные медсправки можно хранить на удаленном сервере базы данных. Благодаря третьему преимуществу врач, не выходя из кабинета, с помощью компьютера (который имеет доступ к медицинской информационной системе), может просмотреть необходимые ему справки, что поможет врачу сэкономить его рабочее время. Каждый производитель медицинского оборудования производят свои аппараты по анализу крови, следовательно, каждый прибор будет выдавать свою уникальную электронную медсправку по анализу крови.

Поэтому целью моей работы являлось написание программы, которая будет все эти медсправки приводить к единому виду (стандарту). А именно, доставать необходимые ключевые данные пациента (исследование, результат, размерность результата), которые в свою очередь сохраняются в виде отдельных файлах с расширениями .txt и .csv.

Язык программирования Python

Python — активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года [2].

Данный язык программирования - это мощный инструмент для создания программ самого разнообразного назначения, доступный даже для новичков. С его помощью можно решать задачи различных типов. Именно поэтому мое предпочтение остановилось на языке Python.

Анализ ключевых данных из медицинских справок

Перед тем, как приступить к доработке программы, необходимо было вручную проанализировать все ключевые данные, которые используются в медицинских справках. То есть, нам надо было составить обширную базу данных для всех исследований и их размерностей. Во многих медицинских учреждениях России, врачи по-разному указывают названия исследований и их размерностей в медицинских справках. Например, слово «гемоглобин» может писаться как по-русски", так и по-английски «hemoglobin» или может указываться в аббревиатуре Hgb. Точно также дело обстоит и с размерностями. На примере того же гемоглобина размерность может указываться в г/л, в Gm/100ml и т.д.

В результате анализа всех доступных медицинских справок, был составлен excel файл, в котором записаны всевозможные названия исследований крови и их размерности.

Создание файлов для хранения ключевых данных

После создания excel файла с ключевыми медицинскими данными, появился другой немаловажный вопрос: «Где и в каком виде хранить ключевые данные, чтобы программа без проблем могла работать с ними?».

Данные лучше хранить в отдельных файлах, нежели создать массив и хранить их в коде программы. Потому, что если мы создадим массив и запишем туда все наши ключевые данные, то размер файла с программным кодом увеличится в несколько раз. Из-за этого программа будет работать медленнее, да и разработчику труднее будет редактировать и добавлять новые данные в массив.

Было решено хранить данные на веб-сервере в папке blood. Данная папка находится в каталоге \content\data\. Данная структура файлового дерева содержит в себе смысловую нагрузку. А именно: папка content – эта папка, где хранятся все медицинские справки, а также ключевые данные; папка data – эта папка, где могут храниться имена папок, конкретных ключевых медицинских данных (Например, в папке blood хранятся все ключевые данные крови).

Данная реализация файлового дерева обусловлена тем, что данные хранятся на веб-сервере в четком структурированном порядке. Это значит, что

любой разработчик, который будет работать с данной программой, сможет быстро находить нужные файлы.

Было решено разделить названия исследований и размерности друг от друга и хранить в отдельных .dat файлах. Файлы называются measureBlood.dat и varBlood.dat. В measureBlood.dat хранятся названия исследований крови, а в varBlood.dat хранятся размерности крови.

Структура хранения ключевых данных внутри файлах с расширением .dat объясняются следующим образом. Каждое ключевое значение выделяется квадратными скобками и начинается с новой строки. Это делается для того, чтобы в дальнейшем регулярное выражение могло без проблем вытаскивать все значения и записывать их в массив.

Регулярные выражения

Регулярные выражения — это своеобразный фильтр для текстовых данных. Например, нужно найти все doc-файлы на съемном носителе. Вручную искать долго и непродуктивно. Достаточно в поисковой строке ввести текст «*.doc», и система отберет все файлы с любым именем формата .doc [3].

Загрузка ключевых данных в массив

Программе (во время работы с медицинскими данными) необходимо иметь загруженные ключевые данные (из файлов с расширением .dat) в массиве. В дальнейшем, с помощью этого массива, программа будет проводить сравнительный анализ ключевых данных с медицинской справкой. А также вытаскивать нужные данные с помощью регулярных выражений.

Функция open_read_measureBlood полностью считывает файл measureBlood.dat. А потом с помощью регулярного выражения загружает нужные данные в массив measureBlood. Данная функция загружает размерности крови.

Функция open_read_varBlood полностью считывает файл varBlood.dat. А потом с помощью регулярного выражения загружает нужные данные в массив varBlood. Данный функция загружает названия исследований крови.

Извлечение необходимых данных из медсправки

После загрузки ключевых данных в массивы measureBlood, varBlood, а также загрузки исходной справки в переменную content, необходимо извлечь нужные данные из content и записать их в переменные matches_list и res.

Функция extract_data_from_blood с помощью регулярного выражения извлекает из медицинской справки названия исследований, их числовые значения и записывает полученные данные в массив matches_list.

Функция extract_measure_blood проводит сравнительный анализ медицинской справки с массивом measureBlood. Если функция обнаруживает совпадение размерностей между массивом и справкой, то функция записывает совпадающую размерность в массив res.

Функция concatenation_result объединяет массивы matches_list и res в единое целое. После соединения массивов, функция сохраняет полученные общие данные справки в отдельные файлы result.txt и result.csv.

Результат работы программы представлен на рисунке 1.

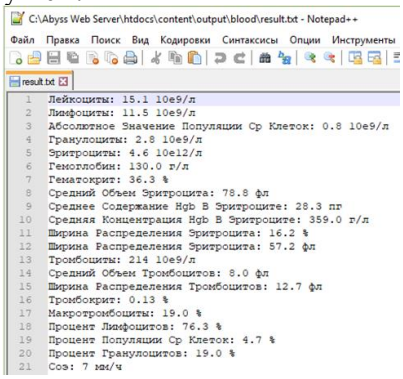

Рис. 1. Необходимые извлеченные данные из медицинской справки

Заключение

Данная программа производит извлечение ключевых слов (исследования крови), числовые значения и размерности из медицинских справок. Обработанные данные сохраняться в виде отдельных файлах с расширениями .txt и .csv. В дальнейшем сохраненные данные будут сохраняться в БД медицинской информационной системы.

Список использованных источников

1. Когаловский М. Р. Перспективные технологии информационных систем. — М.: ДМК Пресс; Компания АйТи, 2003. — 288 с.
2. Язык программирования Python 3 [Электронный ресурс] / Python 3 для начинающих. – URL: https://pythonworld.ru/ (дата обращения: 17.11.2018)
3. SEMANTICA [Электронный ресурс] / Что такое регулярные выражения – URL: https://semantica.in/blog/chto-takoe-regulyarnye-vyrazheniya.html (дата обращения: 19.11.2018)